Pacific University
CommonKnowledge

Volume 10 (2010)

Interface: The Journal of Education, Community and Values

3-1-2010

# My Automated Player Feedback Quagmire

Chris Pruett

# My Automated Player Feedback Quagmire

# My Automated Player Feedback Quagmire

Posted on **March 1, 2010** by **Editor**


**By Chris Pruett**

One of the best ways to develop an idea for a game is to build a prototype and have people play it. Being able to watch somebody play a game, even in a prototype form, is extremely valuable. Almost immediately rough spots in the design become apparent; there will be details that players miss, or controls that players have trouble with, or they will discover a method of playing that isn't what the creator had in mind. The very best games are often those that incorporate user testing from a very early stage of development, and use that feedback to tune and modify the design of the game throughout development.

When I worked at a game company, we would often bring children (our target audience was usually 8 to 12 year olds) from the local community in to test our games. Kids are great testers because they don't get frustrated easily and have a higher tolerance for flawed games than adults. We would bring the kids in, hook the game machine up to a VCR, and record the whole play session. Usually the designer would sit in with the kids and talk to them about their play experiences; hearing what they have to say as they play can often be as valuable as the video of the play session itself. After the kid testers have left, the designer would pore over the video recordings, trying to identify areas that were problematic.

The basic goal of these tests is usually to identify frustration points. Frustration is the enemy of fun, and it is very hard to know what players will find frustrating without doing real world tests. Frustration can be the result of the game giving the player unclear goals, or being too difficult, or otherwise arresting the players' progress. Often, frustration happens because some message that the designer intended to communicate to the player does not make it through. A good game is one that is easy to understand yet challenging; a bad game is one that is challenging because it is hard to understand. By watching kids play, a designer can quickly identify these kinds of problems and take steps to resolve them.

For the last year I have been developing a small game in my free time called *Replica Island* [1].

The game is a side-scrolling action game, part of the platformer genre. My day job involves working with Google's Android mobile operating system, and Replica Island is my first Android-based game. Though the game started out as a technical exercise, over the last six months I have been almost totally focused on the design of the game itself; though the basic technology has been in a completed state for a long time, developing the content of the game has proof to be a significant challenge.

Part of the challenge is that I have no way to test my designs on other people. I do not have a local community of kids who are willing to volunteer their time (nor are kids my target audience for this title). I do not have a way to connect my game to a VCR and record a play session; most Android devices are cell phones without any sort of video connectors. And finally, even if I do get somebody to play the game, it is hard to see what sort of mistakes they are making without hovering right over their shoulder. So while I know from experience that user testing is really the only way to verify the quality of any given game design, the logistics of testing on my chosen platform are problematic.

After some deliberation, I decided to try a different approach. Rather than record live play sessions, my game now reports statistics about players back to a server, which I can then use to determine which areas are particularly problematic. I released a beta version of my game internally to other Google employees, and several hundred were nice enough to give it a spin. The data I collected from those users has proven extremely useful in tuning the design of the game and identifying frustration spots, and I'm very pleased with the results [2].

The problem with this approach is that it requires walking a fine line between privacy concerns and test integrity. In order for the data to be reliable, the player should ideally not know that he is being watched. Even though I only send back a very small amount of information (notifications are sent when a player "dies" or completes a stage), being observed causes some players to change their behavior. Ive experienced this myself; I find myself playing very differently in games that I know are keeping an internal score card of my achievements compared to those that do not. So in the interests of data integrity, I would rather that my players not be aware of my data recording system.

But on the other hand, any time I send data about a specific person back to a server, I am increasing the potential for somebody to complain about privacy. The average user cannot tell what data I am sending, or how I am sending it, or whether that data could be used to build some sort of personal profile about them. In reality, I have been careful not to include any personal details in the data I send. Each player is assigned a random number and I have no way to tie that number to any sort of personal information. I have been pretty careful to ensure that this data is really just game-related, and even if some nefarious third party got their hands on it, it would not be possible to identify users with it. But the user cannot know that I have taken such precautions–it is natural for an internet-savvy player to wonder a bit about why my game needs an internet connection and what kind of data it is transmitting.

Of course, for the purposes of my test I have only released the game to fellow employees, so there is no real trust problem. But as Replica Island nears completion, I am very strongly inclined to include the data collection system in the final version that goes out to average joe users. With this system operating in the wild, I would be able to collect huge amounts of play data and make adjustments to the game even after it is released. Thats a really attractive idea; I would love to be able to continue to improve the play experience as my understanding of my players grows.

But, then again, maybe it is not worth the confusion and suspicion that comes with transmitting user data back home. Android lets users know which apps use the internet, and there is no other internet-related functionality in my game. It would be a shame if the very system I included to improve the game caused some players to refuse to install it.

I will probably choose a safe middle ground by letting players opt-out of the data collection system via a setting screen. This is not an ideal solution because it lets everybody know that there is a data collection system in the first place, and some players' data will subsequently be influenced by that knowledge. But it is probably a better alternative than trying to be sneaky about it or cutting the system out altogether. Letting the user make an informed decision is probably the best way to go.

At any rate, I plan to release the source code for Replica Island along with the game, so anybody who is interested can pick it apart and see how it works. If I do ship the game with the data collection system enabled, my next challenge will be actually interpreting all of the automated feedback that I receive (and, if the title is popular, keeping my server from crashing). So far, my tests have been limited to a few hundred users; I do not know what will happen when that is expanded to 1000 or 10000 or 50000 users. Still, that is a nice problem to have. If 50000 people decide to play my game, I must be doing something right.

### Endnotes

[1] For more information, see http://www.replicaisland.net

[2] More information about the system I built for Replica Island is detailed on my development blog: http://replicaisland.blogspot.com/2010/01/tuning-with-metrics-redux.html

This entry was posted in Uncategorized by **Editor**. Bookmark the **permalink [http://bcis.pacificu.edu/interface/?p=3754]** .

3 THOUGHTS ON "MY AUTOMATED PLAYER FEEDBACK QUAGMIRE"

**Editor**
on **March 1, 2010 at 11:41 AM** said:

Poster Name: michael
Message: Chris, I can't seem to be able to move the possession orb on my G1. What's the problem?

**Wilford**
on **January 31, 2014 at 4:59 PM** said:

Nice post. I learn something new and challenging on
blogs I stumbleupon on a daily basis. It's always exciting to read articles from other authors and practice a little something frpm other web sites.

**Odell Zan**
on **February 6, 2014 at 3:34 AM** said:

Hallo an alle, hier wie Ashley sind wir alle?